Protocollo a chiave pubblica: Elgamal

Crittografia

Luciano Margara

Unibo

2025

Introduzione

Elgamal è un sistema di cifratura a chiave pubblica, proposto dal ricercatore egiziano-americano Taher Elgamal nel 1985. Lo schema è basato sulla difficoltà del calcolo del logaritmo discreto.

Dati x, y, n calcolare z che soddisfi
$y = x^z \mod n$

Campo finito

Un campo finito è una struttura algebraica $(F, +, \cdot)$ dove:

1. F è un insieme non vuoto.
2. Le operazioni di addizione $(+)$ e moltiplicazione $(\cdot)$ sono definite su F.
3. Le operazioni soddisfano le seguenti proprietà:
   - Associatività e Commutatività dell'addizione.
   - Esistenza dell'elemento neutro e dell'opposto dell'addizione.
   - Associatività e Commutatività della moltiplicazione.
   - Esistenza dell'elemento neutro e dell'inverso della moltiplicazione (tranne elemento nullo).
   - Distributività: $\forall a, b, c \in F : a \cdot (b + c) = (a \cdot b) + (a \cdot c)$.
4. Il campo ha un numero finito di elementi, spesso denotato come $|F|$, che corrisponde al suo ordine.

l'ordine di un campo finito (la sua cardinalitá) è sempre una potenza di un numero primo

Esempio di campo finito

Un esempio di campo finito è il campo finito $\mathbb{F}_p$, dove p è un numero primo. Ad esempio, se $p = 5$, allora il campo finito $\mathbb{F}_5$ consiste nei numeri $\{0, 1, 2, 3, 4\}$ con le operazioni di addizione e moltiplicazione eseguite modulo 5. Questo campo finito è denotato come $\mathbb{F}_5$ e rappresenta un esempio semplice ma significativo di un campo finito.

Generatore di un campo finito

Un elemento g di un campo finito F è detto generatore se ogni elemento non nullo di F può essere espresso come potenza di g. Formalmente, g è un generatore se $F \setminus \{0\} = \{g^0, g^1, g^2, \dots, g^{|F|-2}\}$. Ad esempio in $\mathbb{F}_p$ gli elementi $2, \dots, p-1$ sono generatori.

Generatore di un campo finito

Consideriamo ad esempio $\mathbb{F}_7$. L'elemento 3 è un generatore perché:

$3^0 = 1$ (mod 7)

$3^1 = 3$ (mod 7)

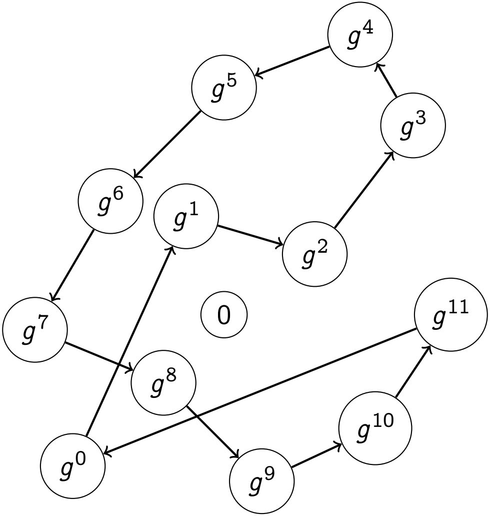
$3^2 = 2$ (mod 7)

$3^3 = 6$ (mod 7)

$3^4 = 4$ (mod 7)

$3^5 = 5$ (mod 7)

Generatore di un campo finito



Campo da gioco

Ogni campo finito ha p^n elementi, per qualche numero primo
p e qualche numero naturale n>=1.
Quando il campo finito ha esattamente p elementi (n=1) le sue
operazioni vengono definite tramite l'aritmetica modulare modulo p.

Sia F un campo finito con q elementi

Sia $g \in F$ un generatore di F

Generazione delle chiavi

A e B generano una coppia di chiavi ciascuno

$$\begin{cases} Prv_A = (\text{numero scelto a caso nell'insieme } \{1, \ldots, q-1\}) \\ Pub_A = g^{Prv_A} \; mod \; q \end{cases}$$

$$\begin{cases} Prv_B = (\text{numero scelto a caso nell'insieme } \{1, \ldots, q-1\}) \\ Pub_B = g^{Prv_B} \; mod \; q \end{cases}$$

Questi passaggi assicurano che, pur mantenendo segrete le chiavi private, le chiavi pubbliche possano essere condivise e utilizzate per calcolare una chiave condivisa in fase di cifratura.

Codifica

Quando A vuole inviare un messaggio m a B

A codifica e spedisce un messaggio $m < q$ a B

A genera un numero random $k \in \{1, \ldots, q-1\}$

Questo valore è detto effimero perché viene utilizzato solo per quella sessione di cifratura e rende ogni messaggio cifrato diverso anche se il messaggio originale fosse identico.

A calcola:

Calcolo della Chiave Condivisa che solo B (che conosce PrvB) potrà ricostruire.

$$K_A = Pub_B^k \ mod \ q$$

$$C_1 = g^k \ mod \ q$$

Questa parte trasmette indirettamente il valore k (senza rivelarlo) e sarà utilizzata da B per ottenere la stessa chiave condivisa.

$$C_2 = K_A m \ mod \ q$$

Qui il messaggio m viene "offuscato" moltiplicandolo per la chiave condivisa.

A spedisce a B la coppia (C_1, C_2)

Deodifica

B decodifica un messaggio ricevuto da B

B riceve (C_1, C_2)

B calcola:

$$K_B = C_1^{Prv_B} \ mod \ q$$

Ricostruzione della Chiave Condivisa: B usa la sua chiave privata per calcolare KB e si nota che KB coincide con KA calcolato da A, grazie alle proprietà esponenziali:
KB = C1^PrvB modq = (g^k)^PrvB = g^(k*PrvB) mod q

$$m = C_2 K_B^{-1} \ mod \ q$$

Conoscendo KB, B calcola l'inverso moltiplicativo KB^−1 in F (garantito dall'essere un campo finito) e ottiene il messaggio originario.

Correttezza

Dimostriamo che $K_A = K_B$

$$
\begin{aligned}
K_A &= Pub_B^k \ mod \ q \\
&= (g^{Prv_B} \ mod \ q)^k \ mod \ q \\
&= g^{kPrv_B} \ mod \ q \\
K_B &= C_1^{Prv_B} \ mod \ q \\
&= (g^k \ mod \ q)^{Prv_B} \ mod \ q \\
&= g^{kPrv_B} \ mod \ q
\end{aligned}
$$

Correttezza

... e quindi

$$
\begin{aligned}
C_2 K_B^{-1} \; mod \; q &= (K_A m \; mod \; q) K_B^{-1} \; mod \; q \\
&= K_A m K_B^{-1} \; mod \; q \\
K_A = K_B \implies \quad &= m \; mod \; q \\
&= m
\end{aligned}
$$

Semplificando

Mitt	Intruso	Dest
messaggio $= m$		$Prv = R \qquad Pub = g^R$
$g^R \leftarrow \cdots$	$\cdots \leftarrow g^R$	
genera k a caso		
	C1 $g^k \rightarrow \cdots$	$\cdots \rightarrow g^k$
	C2 $m(g^R)^k \rightarrow \cdots$	$\cdots \rightarrow m(g^R)^k$
		$m\underbrace{(g^R)^k((g^k)^R)^{-1}}_{\text{si semplificano}} = m$

Sicurezza

- Un attaccante non può ricavare PrvB a meno di risolvere il problema del logaritmo discreto (PrvB = log base g (PubB) mod q).
- Non può calcolare la chiave condivisa KB senza risolvere il problema del Diffie-Hellman computazionale (KB = g^(k*PrvB) mod q).

Per questo motivo, il sistema ElGamal è considerato sicuro per cifrare i messaggi, a meno che non si usino valori di q troppo piccoli, che renderebbero fattibile un attacco con forza bruta o con metodi avanzati di fattorizzazione.

L'intruso conosce: q, g e $Pub_B = g^{Prv_B} \ mod \ q$

vede passare sul canale:

$C_1 = g^k \ mod \ q$

$C_2 = {Pub_B}^k m \ mod \ q$

e vorrebbe calcolare:

m o ancora meglio Prv_B